UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a master's thesis by

Penghuan Ni

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Name of Faculty Advisor

Signature of Faculty Advisor

Date

GRADUATE SCHOOL

Your Great Title For Your Thesis

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Penghuan Ni

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Professor Advisor Name

December 2016

Acknowledgements

Some acks are nice.

Dedication

Your dedication goes here.

Abstract

In the past century, there are thousands of cities have been built around the world. At the same time, more and more city environmental drawbacks have been came out, such as, heat island effect, air pollution problem, etc. In our research, we are trying to apply Nondominated Sorting Genetic Algorithm II(NSGA-II) to find the Pareto Optimal Solutions of Urban Design Optimization Problem. In addition, since NSGA-II especially gives a good performance on solving dynamic variables involved problem, we are interested to see how well it could deal with dynamic variables(e.g., wind, sunshine, temperature, etc.) in Urban Design Optimization Problem. To solve this problem, we implement and modify NSGA-II and expect that it could give us multiple Pareto Optimal Solutions and help us to make a better decision.

Contents

List of Figures

1 Introduction

In this paper, we applied the NSGA-II to Urban Design Model to find the Pareto-optimal solutions. Urban design is a complicated problem, it involved many factors, such as sunshine, wind, water area, building height, street width, trees, public space, etc. How do we take into account all those factors becomes a worldwide problem. In the past, people could only make decisions based on their past experience. It turns out that experience could give us some decent results, however, it will never always give us the best result. Human beings are limited and always make mistakes. In this Computer Age, why don't we seek help from computer. Computer has a lot of advantages compared with human being, it is more reliable, its computation ability is much more than human being. We could simulate this problem with computer and ask computer to solve it for us.

2 Background

2.1 Background

Since first computer was invented in 1940s, computer has became a part of our daily life. At the beginning, the purpose of computer is just to help people do some complicated computing. After Internet was invented in 1960s, our human being has entered into a new age. With more than half century's developing, computer and Internet have much more power than they were first invented. Artificial Intelligence, Big Data, Cloud Computing, more and more new computer technology have been applied into our life. All those technologies make our life better and easier. We can use our smartphone to control our house electric appliance no matter where we are. Computer could simulate the energy we used at home and give us solutions for building house to have a lower energy consumption. Super market could analysis the users' shopping data to re-arrange the products' place to improve the sales volume. New technologies has made an influence in every aspect of our life. Artificial intelligence is one of those new technologies which could make our life better.

Machine Learning, which is a big part of Artificial intelligence, is about learning from the experience and predicating for the future situation or solving some new problems. Self-driving cars, practical speech recognition, effective web search, face identification, and so many new technologies have been developed by using machine learning. Everyone probably use machine learning dozens of times a day without knowing it. There are many approaches in machine learning and different approaches have their own benefits and drawbacks. Genetic algorithm is a widely used algorithm to help people find solutions for some problems.

From genetic algorithm was first introduced in 1975 until now, it has been well developed and applied in many area. In the following parts of this chapter, we will introduce some versions of genetic algorithm and also NSGA-II, which we will use for our own problem. Different genetic algorithm may fit with different problems. Based on the previous work of people using NSGA-II on some multi-objective optimization problems[8], NSGA-II could also apply for Urban Design optimization problem.

2.2 Previous Work

The idea of Genetic Algorithm(GA) was from Darwin's Theory of Evolution and was first invented by John Henry Holland in 1975[7]. Mimicking the process of natural selection, GA uses selection, crossover, and mutation operations to generate solutions and tries to find the optimal solution. More specifically, an initial random population is given at the beginning. Then based on criterions or objectives, each individual is assigned a fitness value. After selecting a certain number of individuals from the initial population based on their fitness values, crossover operation with a crossover probability is applied on those individuals to generate some new offspring. Next, mutation operation with a mutation probability is applied on the new offspring. After all those steps, hopefully it can give us a new population which have some individuals with better fitness values. Last, consider this output population as the new initial population and repeat selection, crossover and mutation operations until pareto optimal solutions are achieved. After this framework was constructed, it has been used in many areas, such as Artificial Intelligence, Bioinformatics, Building Design, Scheduling Problems, etc. While, on the way to solve practical problems, people have put a lot of effort to improve GA. Next, some of those milestone variations of GA will be shown.

2.2.1 Multiobjective Genetic Algorithm(MOGA)

In general, optimal problems could be divide into two categories based on the number of objectives: Single Objective problems and Multi-Objective problems. In order to understand MOGA, a toy example is necessary. A simple example of a single objective problem is to find the minimum value of $y = x^2$ with $x \in [-3, 3]$. Clearly, the minimun value could be achieved at $x = 0$, $y = 0$. However, when there are two objectives, the problem becomes more complicated. For example, instead of just minimizing $y = x^2$, both $y_1 = x^2$ and $y_2 = (x - 2)^2$ want to be minimized. Then, there is not a specific x value for this problem. Figure 2.1 shows us x at 0 could achieve minimum value of $y_1 = x^2$ and x at 2 could achieve minimum value of $y_2 = (x - 2)^2$. However, there is no x value which could guarantee both y_1 and y_2 at their minimun value. For $x = 0$ or 2, people usually call them Pareto Optimal Solutions[6][11]. Other than $x = 0$ or 2, any points between 0 and 2 could be a valid compromise solution.

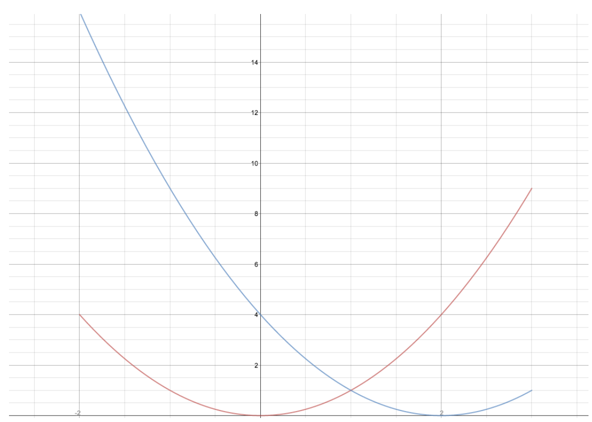


Figure 2.1: Functions $y_1 = x^2$ and $y_2 = (x - 2)^2$

In the real world, there are many kinds of simultaneous optimization of multiple objectives. People have tried many methods to find Pareto Optimal Solution. The classical method is to convert a multiobjective problem to a single objective problem. In paper [10], three classical methods are introduced: 1.Objective Weighting, 2.Distance Functions, and

3.Min-Max Formulation. All those methods have one common drawback: they can only give us a single-point solution. However, when people make decisions, they often need different alternatives. Another significant drawback of those methods is they heavily depend on what weight vector or demand level chosen, which requires that the user have knowledge about the underlying problem. However, in most cases, users do not have much knowledge about the problem and that is one of the reasons they use genetic algorithm. So, those methods are not efficient and users probably need to try many times to get an acceptable solution.

Since classical methods are neither adequate nor convenient on solving multiobjective optimization problems, people started to find other ways to implement GA.

2.2.2 Vector Evaluated Genetic Algorithm(VEGA)

VEGA is the first practical algorithm and was developed by Schaffer in 1984[9]. Instead of changing a multiple objective problem to a single objective problem, Schaffer modified the simple traditional genetic algorithm by adding independent selection cycles according to every objective function[9]. Bascially, to fill up a portion of the mating pool, he created a loop around the traditional selection procedure to repeat selection method for each individual objective. After this selection step, we could have a thoroughly shuffled population to apply crossover and mutation steps. This could make the mating of individuals from different subpopulation groups possible.

With the independent selection of specialists, we could make the population in speciation. After a large number of generations, we could see the convergence of the entire population toward the individual optimum regions. To make a decision, we may not want to have any bias on such middling individuals, rather, we desire more nondominated points. Schaffer developed two heuristics to minimize this speciation: the nondominated selection heuristic and the mate selection heuristic. For the nondominated selection heuristic, he penalized the dominated individuals by substracting a small fixed penalty from their expected number of copies druing selection. Then the total penalty for dominated individuals was divided among the nondominated individuals and was added to their expected number of copies during selection. For the mate selection heuristic, he wanted to promoted the crossbreeding of specialists from different subgroups by selecting an individual, as a mate to a randomly selected individual, which has the maximum Euclidean distance in the performance space from its mate. However, both of those two heuristics are failed due to some certain problems.

Overall, VEGA could solve the drawback in the classical MOGA and give us multiple Pareto Optimal Solutions without depending on the weight or demand level. But, in some cases it suffered from its bias toward some individuals or regions. After that, people tried many other methods to avoid the bias.

2.2.3 Nondominated Sorting Genetic Algorithm(NSGA)

NSGA was first introduced by N. Srinivas and Kalyanmoy Deb in 1994[10]. They investigated Goldberg's notion of nondominated sorting in GAs along with a niche and speciation method to find multiple Pareto Optimal Solutions simultanesously. The reason they call this algorithm the Nondominated Sorting Genetic Algorithm is that it build on a nondominated sorting procedure. Two ideas behind the nondominated sorting procedure are ranking selection which is used to emphasize good points method and niche method which is used to maintain stable subpopulations of good points.

The only difference between simple genetic algorithm and NSGA is the way selection operator works. First, we rank the population based on the individuals' nondomination and identify those nondominated individuals from the current population. Second, let all those individuals constitute the first nondominated front and assign a large dummy fitness value to them. All those nondominated individuals which have an equal reproductive potential should have same fitness value. Third, to maintain diversity in the population, we apply sharing methods[4][3] on these classified individuals. Sharing is achieved by performing selection operation using degraded fitness values that are obtained by dividing the original fitness value of an individual by a quantity proportional to the number of individuals around it. Last, ignore those nondominated multiple optimal points and using the same method to process the rest of population to get the second nondominated front. This time, we assign a new dummy fitness value to the second nondominated front which is smaller than the first nondominated front. By this process, we could classified the entire population into several fronts. After the classification process, we reproduct the population by each individual's dummy fitness value and apply a stochastic remainder proportionate seletion on them.

After this selection process, we move to crossover operation. Since we want to find nondominated regions or Pareto Optimal fronts and first front have the maximum fitness value, we give them more copies. This could give us a fast convergence of the population toward nondominated regions. Furthermore, sharing method could help us to distribute the population over this region. By nondominated sorting procedure, NSGA becomes faster than VEGA. And NSGA can solve both minimization and maximization problems. However, it is still not very convenient. We still need to specify the sharing parameter and it is lack of elitism.

2.2.4 Nondominated Sorting Genetic Algorithm II(NSGA-II)

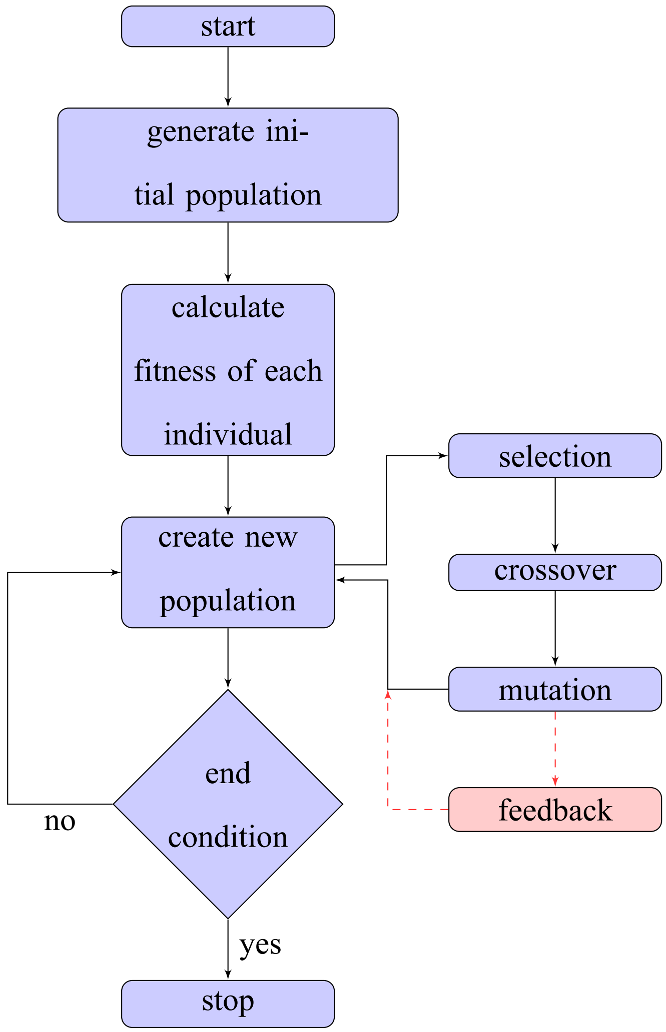
In 2002, Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan made some modification on NSGA to upgrade it to NSGA-II[2]. After NSGA-II was introduced, it has been used in many area, such as Building Design[8], Hydro-Thermal Power Scheduling[5], Soil and Water Assessment Tool[1], etc. Moreover, in our paper, we will use it to solve Urban Design problem.

Compare with the previous NSGA, NSGA-II has improved the computational efficiency, elitism, and sharing parameter[2]. For computational complexity of nondominated sorting, it improve from $O(MN^3)$ to $O(MN^2)$ by using fast nondominated sorting approach[2]. With the elitism, NSGA-II could not only speed up the GA performance, but also prevent loss of good solutions. Moreover, NSGA II does not need to specify a sharing parameter σ_{share}, which is a requirement for pervious multi-objective evolutionary algorithms. Instead of using sharing function approach, NSGA-II is using Crowded-Comparison Operator, which could both improve the computational complexity and maintation a spread of solutions.

3 Implementation

3.1 Steerable GA Explanation

Our Steerable Genetic Algorithm take user's feedbacks into account and generate an user preferred solution. Following is the flowchart for our Steerable GA and there are more explanations under the flowchart.


start
generate initial population
calculate fitness of each individual
create new population
selection
crossover
mutation
feedback
end condition
no
yes
stop

3.1.1 Run and Compile

3.1.2 Input and Output

3.1.3 Operators

3.2 New Section For Next Important Topic

3.2.1 Algorithm Initialization

3.2.2 Atomic Operations

You may even need code in your thesis. Here is a way to nicely include code with LaTeXusing the listings package.

```
for (unsigned int idx=0; idx<maxSize; idx++) {
  atomic_add( idx );
}
```

3.2.3 Programming Style

Explaining Fine Detail Here

TODO: Make sure to finish this!

Last Subsection

4 Results

Your results. This worked great. Here's a plot to show how great it worked.

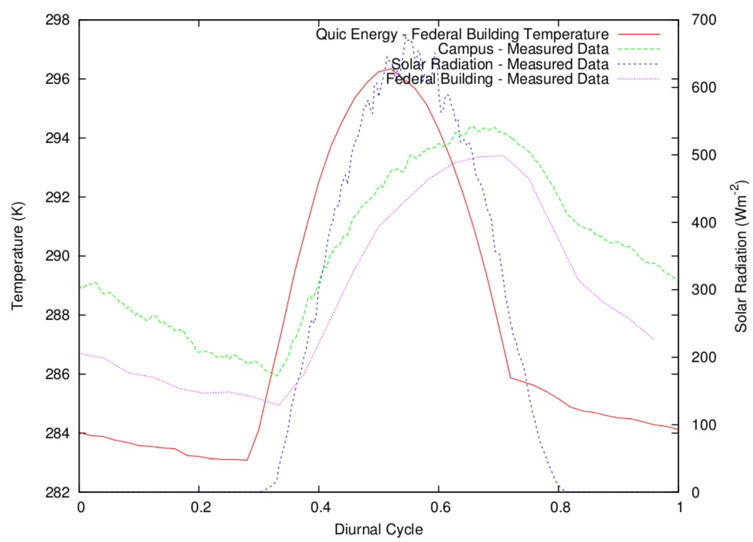


Figure 4.1: Good data.

Figure 4.1

5 Conclusions

How can you wrap this up?

A Appendix A

Do you need an Appendix? You can include several of them if you want.

Bibliography

[1] E. G. Bekele and J. W. Nicklow. "Multi-objective automatic calibration of SWAT using NSGA-II". In: *Journal of Hydrology* 341.3 (2007), pp. 165–176 (cit. on p. 8).

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *Evolutionary Computation, IEEE Transactions on* 6.2 (Apr. 2002), pp. 182–197. ISSN: 1089-778X. DOI: 10.1109/4235.996017 (cit. on pp. 7, 8).

[3] K. Deb. *Genetic algorithms in multimodal function optimization*. Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama, 1989 (cit. on p. 7).

[4] K. Deb and D. E. Goldberg. "An investigation of niche and species formation in genetic function optimization". In: *Proceedings of the 3rd international conference on genetic algorithms*. Morgan Kaufmann Publishers Inc. 1989, pp. 42–50 (cit. on p. 7).

[5] K. Deb, S. Karthik, et al. "Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling". In: *Evolutionary Multi-Criterion Optimization*. Springer. 2007, pp. 803–817 (cit. on p. 7).

[6] A. Hans. "Multicriteria optimization for highly accurate systems". In: *Multicriteria Optimization in Engineering and Sciences* 19 (1988), pp. 309–352 (cit. on p. 4).

[7] J. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975. ISBN: 9780472084609. URL: https://books.google.com/books?id=YE5RAAAAMAAJ (cit. on p. 3).

[8] L. Magnier and F. Haghighat. "Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network". In: *Building and Environment* 45.3 (2010), pp. 739–746 (cit. on pp. 3, 7).

[9] J. D. Schaffer. "Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms (Artificial Intelligence, Optimization, Adaptation, Pattern Recognition)". AAI8522492. PhD thesis. Nashville, TN, USA, 1984 (cit. on p. 5).

[10] N. Srinivas and K. Deb. "Muiltiobjective optimization using nondominated sorting in genetic algorithms". In: *Evolutionary computation* 2.3 (1994), pp. 221–248 (cit. on pp. 4, 6).

[11] C. Vira and Y. Y. Haimes. *Multiobjective decision making: theory and methodology*. 8. North-Holland, 1983 (cit. on p. 4).